

PATENT APPLICATION

CLB9-B95

**METHOD AND SYSTEM FOR DYNAMICALLY MANAGING WIRELESS
COMMUNICATIONS SYSTEM PROTOCOLS**

363027-4630260

Inventors: T. Allan Hamilton, Michael R. Watson, Ray Chock, Sam Gebala
San Jose, CA; U.S.A.

METHOD AND SYSTEM FOR DYNAMICALLY MANAGING WIRELESS COMMUNICATIONS SYSTEM PROTOCOLS

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates generally to wireless communications between electronic appliances and, more specifically, to a Method and System for Dynamically
5 Managing Wireless Communications System Protocols.

2. Description of Related Art

As manufacturing capabilities have improved, electronic appliances have become smaller and smaller, while simultaneously offering more features and capabilities
10 than their predecessors, and furthermore at reduced cost to the consumer. As a result, portable electronic appliances, such as computers, personal digital assistants (PDA's) and cameras have nearly become commonplace for businesses and even for the casual user. The proliferation of these powerful portable devices has led to the desire for the ability to share and otherwise transfer information without the need for a hardwired connection. Wireless
15 communications technology, and particularly Infrared (Ir) and Radio Frequency (RF) communications between portable electronic appliances has evolved quickly to satisfy this

demand. In fact, it is virtually impossible to purchase either a PDA or Portable Computer that does not include Ir communications capability.

As the physical implementation of wireless communications has expanded rapidly, this rapid development has also resulted in the nearly uncontrolled release of software to interface with the wireless communications hardware. Moreover, it is fairly common for individuals to purchase a particular Ir-enabled appliance (for example) with a pre-installed Ir Communications Protocol Stack (Stack) and Applications Set (App Set) that are obsolete. While the obsolete software is functional, it's performance is not optimal. Furthermore, heretodate it has been impossible to upgrade the App Set because the obsolete Stack would not be capable of communicating with it. Also, even if you could upgrade both the Stack and App Set, you would be incapable of "downgrading" the Stack and App Set in the eventuality that the device with which you wish to link still has its default Stack and App Set.

What is needed, therefore, is a system and method internal to the electronic device that will automatically detect and enable the most advanced Stack and App Set available on both devices.

SUMMARY OF THE INVENTION

In light of the aforementioned problems associated with the prior methods and systems, it is an object of the present invention to provide a Method and System for Dynamically Managing Wireless Communications System Protocols. The preferred method
5 and system will automatically detect and enable the most advanced Stack and Application Set available on a pair of devices in wireless communications with one another. It is an object to provide a device having detector means in wireless communication with another device or devices for detecting the configuration of the application set of the other device(s). The preferred device will further include selector means for enabling the optimum communication
10 protocol stack responsive to the application set configuration of the other device. The preferred device will further enable the optimum internal application set responsive to the application set of the other device. It is a further object that the device begin (and re-establish broken) communications using a default communication protocol stack and application set.

BRIEF DESCRIPTION OF THE DRAWINGS

The objects and features of the present invention, which are believed to be novel, are set forth with particularity in the appended claims. The present invention, both as to its organization and manner of operation, together with further objects and advantages, may best be understood by reference to the following description, taken in connection with the accompanying drawings, of which:

Figure 1 is a front view of a simulation of a portable computer and a printer engaged in infrared communications;

Figure 2 is a depiction of the interaction of the Stack and Applications Set in a pair of conventional Ir-enabled appliances having similar software versions;

Figure 3 is a depiction of the ramification when two Ir-enabled appliances have dissimilar software versions;

Figure 4 is the method of the present invention for optimizing the Stack and Application Set of an electronic appliance having wireless communications capability; and

Figures 5A, 5B and 5C depict the elements of the system of the present invention.

DETAILED DESCRIPTION
OF THE PREFERRED EMBODIMENTS

The following description is provided to enable any person skilled in the art to make and use the invention and sets forth the best modes contemplated by the inventor of carrying out his invention. Various modifications, however, will remain readily apparent to those skilled in the art, since the generic principles of the present invention have been defined herein specifically to provide a Method and System for Dynamically Managing Wireless Communications System Protocols.

The present invention can best be understood by initial consideration of Figure 1. Figure 1 is a front view of a simulation of a portable computer 10 and a printer 12 engaged in infrared communications 14. It should be understood that the present invention also relates to RF and other types of short-range wireless communications system; Ir will be used in the following discussion for the sake of simplicity. It should also be understood that the computer 10 and printer 12 are only exemplary – they represent any of a variety of Ir-enabled electronic appliances. In this example, the portable computer 10 and the printer 12 each include an Ir transceiver 16 and 18, respectively, to enable the infrared communications 14 between the pair of devices.

Now turning to Figure 2, we will delve deeper into the operations of a conventional Ir-enabled appliance. Figure 2 is a depiction of the interaction of the Ir communications-related software packages in the computer 10 and printer 12 of Figure 1. In order for the computer 10 to execute a simple print function via the Ir transceiver 16, the

* (print) application set 20 must communicate with the Ir transceiver 16 through the Ir
communications protocol stack 22. The Ir communications stack 22 is essentially a set of
instructions located in resident memory, that coordinate or translate instructions emanating
* from, and coming into, the application set 20. The application set 20 is one or more software
5 programs residing in permanent and/or resident memory within the computer 10, in this case
for commanding and receiving responses from the printer 12.

Similarly, the printer 12 has its own Ir stack 24 and app set 26. In this
depiction, the computer's app set 20 has a certain identification and version; ID=X, Ver=0 for
example. The stack 22 has its own version or revision; ID=X, for example. As shown, the
10 printer 12 has an app set 26 having ID=X, Ver=0 and a stack 24 with ID=X – since the stacks
22 and 24 and app sets 20 and 26 are of like versions and identifications, the app sets 20 and
26 can interface to permit Ir communications 14.

Now turning to Figure 3, we will analyze the effect of disparities between app
sets and/or stacks. In Figure 3, the computer 10 has installed within it an ID=Y stack 22A
15 and a Y/1 app set 20A. Let us assume that these software packages have been upgraded to
more recent versions by the user because additional functionality was desired. Unfortunately,
the printer 12 has not been upgraded, and the stack 24 and app set 26 are the default ID's and
versions. Now, when Ir communications 14 are attempted between the two (such as if the
computer 10 was sending a print command to the printer 12), the printer app set 26 is unable
20 to respond correctly to the issued command. The results are unpredictable – the printer may
print normally, or it might print garbled data, or it might simply do nothing. Furthermore, in

another situation, the computer 10 might be the device with the default app set and stack, and the printer 12 the device with upgraded software – the result would very possibly be the same.

The reader should note that we discuss the Ir communications 14 as such in the interest of simplicity – it should be understood that Ir communications 14 may be possible between a certain pair of appliances even if the app sets do not match. However, since the app sets may not “understand” one another, some or all of the desired functionality might be lost or may produce erroneous operations.

We shall now turn to Figure 4 to discuss the operation of the present invention. Figure 4 is a block diagram of a preferred method of the present invention for optimizing the Stack and Application Set of an electronic appliance having wireless communications capability. This method is resident within a particular appliance for *internally* optimizing the app set and stack.

Step 400 indicates that the device will first begin with the default stack being enabled. Upon establishment of communications with another device, the instant device will execute step 410 and query the peripheral device for the ID and/or version of its pertinent application(s) (i.e. the print application(s)). Next, step 420 will branch according to the returned application(s) ID and/or version – those branches shown are for example only.

Let's assume that the peripheral device returns the values ID=X, Ver=0 for its app set. In response, the instant device will execute step 430 and enable stack ID=X. Next, step 440 will be executed to enable application set ID=X, Ver=0.

Should the peripheral device return app set values ID=X, Ver=1 for its app set, the instant device will execute step 450 and enable stack ID=X. Next, step 460 will be executed to enable application set ID=X, Ver=1. Obviously, the instant device must have the aforementioned app set and stack versions available in (typically permanent) memory in order
5 to make steps 430, 440, 450 and 460 possible. After completion of steps 440 and 460, the instant device will execute step 480, thereby re-enabling the default stack and app set.

If the peripheral device returns a default ID/version, the instant device will execute step 470, and refrain from changing the stack and/or app set in use. These steps are provided as the barest outline of the steps involved in optimizing the stack and app set in the
10 instant device. While the method is accurate in portraying the pertinent steps involved in the optimization process, it should be realized that other steps might also be executed. Furthermore, the instant device might have a plurality of ID's and versions of app sets and stacks; in such situations, one branch will exist for each unique app set/stack combination.

Review of Figures 5A – 5C will give further insight into the operation of the
15 system of the present invention. As can be seen, the preferred Ir-enabled device 11 (such as the computer 10 of Figures 1 – 3) has a default application set 20. However, in this embodiment, app set 20A having ID=Y and Ver=1, as well as app set 20B with ID=Y and Ver=2 are resident and available within the device 11. The app sets 20A, B and C make up an app set group 28. Also, in addition to the default stack 22, a stack ID=Y is also available,
20 the two stacks making up the stack group 30.

In communication with the app set group 28 and the stack group 30 is a detector means 32 for monitoring and detecting the composition of the app set group 28, the stack group 30, and the app set of the peripheral (or other) device (once Ir "discovery" occurs). The detector means 32 also has the function of enabling or disabling the appropriate app set. In communication with the detector means 32 is a selector means 34 for enabling and disabling the appropriate stack in response to command from the detector means 32. In Figure 5A, the app set group 28 and stack group 30 are in default conditions, such as immediately preceding Ir communications. The detector means 32 and selector means 34 will preferably comprise software running in resident memory.

Figure 5B depicts the device response when the peripheral device returns app set Y/1 (and stack Y). Upon realization by the detector means 32 of the situation, it will send a signal 36 to the selector means 34 to "enable upgraded stack Y", while the detector means enables app set 20A Y/1. As shown in Figure 5C, once the conversion is completed, operations through the stack 22A and app set 20A can proceed; as discussed earlier in connection with Figure 4, and the detector means will re-enable the default app set 20 and also send a signal 38 to the selector means 34 to re-enable the default stack set 22.

Those skilled in the art will appreciate that various adaptations and modifications of the just-described preferred embodiment can be configured without departing from the scope and spirit of the invention. Therefore, it is to be understood that, within the scope of the appended claims, the invention may be practiced other than as specifically described herein.